Capture-Recapture Abundance Estimation using a Semi-complete Data Likelihood Approach

SUPPLEMENT TO "CAPTURE-RECAPTURE ABUNDANCE ESTIMATION USING A SEMI-COMPLETE DATA LIKELIHOOD APPROACH"

By Ruth King*, Brett T. McClintock†, Darren Kidney‡ and David Borchers‡

*University of Edinburgh**, *NOAA National Marine Mammal Laboratory*† *and University of St Andrews*‡

E-mail: Ruth.King@ed.ac.uk

In this supplement we provide sample JAGS code for model M_h considered in Section 4.1 (Appendix A) and the SECR model considered in Section 4.2 (Appendix B). For each example we provide the model specification component of the JAGS code for the four different model-fitting algorithms: (i) semi-complete data likelihood specifying Jeffreys' prior on N (SCD1) and alternative Poisson-Gamma and Beta-Binomial prior specifications; (ii) semi-complete data likelihood specifying the posterior conditional distribution on $N-n$, induced by Jeffreys' prior on N (SCD2); (iii) super-population complete data likelihood approach of Royle *et al.* (2007) (CD:R) and (iv) super-population complete data likelihood approach of Durban and Elston (2005) (CD:DE).

APPENDIX A: JAGS CODE FOR MODEL M_H

In this appendix we provide sample JAGS code for model M_h considered in Section 4.1.

A.1. First semi-complete data likelihood approach - SCD1. The model component of the JAGS code is provided here for the semi-complete data likelihood approach using the Jeffreys' prior specification for N.

```
model{
  Pi <- 3.14159265359
# Priors:
  alpha ~ dnorm(0.0,0.01)
  tau ~ dgamma(0.01,0.01)
  sigma <- 1/sqrt(tau)

  for (i in 1:n) {
    y[i] ~ dbin(p[i],T)
    logitp[i] ~ dnorm(alpha,tau)
    logit(p[i]) <- logitp[i]
  }

  # Calculate probability of not being observed using Gauss-Hermite quadrature
  # q = number of quadrature points
  # weights and nodes correspond to q quadrature points; entered as data

  for(i in 1:q){
    probi[i] <- 1/sqrt(Pi)*weights[i]*(1/(1+exp(sqrt(2)*sigma*nodes[i]+alpha)))^T
  }
  prob<- sum(probi[])

  # Prior for N: Jeffreys' prior - this is incorporated in the zero trick below
  # in specifying the likelihood term
```

```
  # However a prior distribution is needed to be specified on N
  # Use a discrete Uniform prior so the only influence on the posterior
  # distribution is the upper limit

  n00 ~ dcat(prior[])  # prior = rep(1/(M+1-n),M+1-n); entered as data
  n0 <- n00 - 1
  N <- n + n0

  # Use zero trick for model likelihood
  # Note loggam(N) instead of loggam(N+1) because of Jeffreys' prior for N

  logzeroprob <- loggam(N) - loggam(n0+1) - loggam(n+1) + n0*log(prob)
  lambda <- -logzeroprob + 100000
  dummy ~ dpois(lambda)  # dummy = 0; entered as data
}
```

We also provide the associated JAGS code for two alternative prior specifications on N. In particular we consider (i) $N|\mu \sim Po(\mu)$ with $\mu \sim \Gamma(a,b)$, which is equivalent to $N \sim Negative - Binomial\left(a, \frac{b}{b+1}\right)$ and (ii) $N|\psi \sim Binomial(M,\psi)$ where $\psi \sim Beta(a,b)$, equivalent to $N \sim Beta - Binomial(M,a,b)$. To use these priors, replace the line defining `logzeroprob` with:

```
   # For Negative-Binomial(a,b/(b+1)) prior on N:
   # a and b read in as data - for approximate Jeffreys' prior a = b = 0.000001

   p1 <- b/(1+b)
   logzeroprob <- - loggam(n0+1) - loggam(n+1) + n0*log(prob) +
                  loggam(N+a+1) - loggam(a+1) + a*log(p1) + N*log(1-p1)

  # For Beta-Binomial(M,a,b) prior on N:
  # a and b read in as data  - for approximate Jeffreys' prior a = 0.001, b = 1

   logzeroprob <- - loggam(n0+1) - loggam(n+1) + n0*log(prob) + loggam(M+1) -
                  loggam(M-N+1) + loggam(N+a) + loggam(M-N+b) - loggam(M+a+b) +
                  loggam(a+b) - loggam(a) - loggam(b)
```

Note: using the Negative-Binomial model specification achieved a better ESS and ESS/s than the equivalent Poisson-Gamma hierarchical prior model specification (not provided) where the mean Poisson parameter is treated as an auxiliary variable and imputed within the algorithm. Similarly for the Beta-Binomial model specification, using the explicit probability mass function achieved a better ESS and ESS/s than the equivalent hierarchical prior model specification (not provided) where the Binomial probability parameter is imputed within the MCMC algorithm.

A.2. Second semi-complete data likelihood approach - SCD2. The model component of the JAGS code is provided here for the semi-complete data likelihood approach, specifying the posterior conditional distribution of $N - n$ to be of Negative-Binomial form.

```
model{
  Pi <- 3.14159265359
  # Priors:
  alpha ~ dnorm(0.0,0.01)
  tau ~ dgamma(0.01,0.01)
  sigma <- 1/sqrt(tau)
```

```
  for (i in 1:n) {
    y[i] ~ dbin(p[i],T)
    logitp[i] ~ dnorm(alpha,tau)
    logit(p[i]) <- logitp[i]
  }

  # Posterior conditional distribution for N-n (and hence N):

  n0 ~ dnegbin(pstar,n)
  N <- n + n0

  # Calculate probability of not being observed using Gauss-Hermite quadrature
  # q = number of quadrature points
  # weights and nodes correspond to q quadrature points; entered as data
  for(i in 1:q){
    probi[i] <- 1/sqrt(Pi)*weights[i]*(1/(1+exp(sqrt(2)*sigma*nodes[i]+alpha)))^T
  }
  pstar <- 1-sum(probi[])

  # Use zero trick for initial 1/(pstar)^n

  loglikterm <- -n*log(pstar)
  lambda <- -loglikterm + 100000
  dummy ~ dpois(lambda)  # dummy = 0; entered as data
}
```

A.3. Super-population complete data likelihood approach - CD:R. The model component of the JAGS code for the super-population complete data likelihood approach of Royle *et al.* (2007).

```
model{
  # Priors:
  psi ~ dbeta(0.001,1)
  alpha ~ dnorm(0.0,0.01)
  tau ~ dgamma(0.01,0.01)
  sigma <- 1/sqrt(tau)

  # Complete data likelihood:
  for(i in 1:M){
    y[i] ~ dbin(pi[i],T)
    pi[i] <- z[i]*p[i]
    z[i] ~ dbern(psi)
    logit(p[i]) <- logitp[i]
    logitp[i] ~ dnorm(alpha,tau)
  }

  # Calculate N:
  N <- sum(z[1:M])
}
```

A.4. Super-population complete data likelihood approach - CD:DE. The model component of the JAGS code for the super-population complete data likelihood approach of Durban and Elston (2005).

```
model{
  # Priors:
  alpha ~ dnorm(0.0,0.01)
  tau ~ dgamma(0.01,0.01)
  sigma <- 1/sqrt(tau)

  # Prior for N: (Jeffrey's prior over {n,n+1,...,M} following Link 2013).

  n00 ~ dcat(prior[])  # prior = rep(1/(M+1-n),M+1-n); entered as data
  n0 <- n00 - 1
  N <- n+n0

  # Use zero trick for factorial term
  # Note loggam(N) instead of loggam(N+1) because of Jeffrey's prior for N

  logzeroprob <- loggam(N) - loggam(n0+1) - loggam(n+1)
  lambda <- -logzeroprob + 1000
  dummy ~ dpois(lambda) # dummy = 0; entered as data

  # Complete data likelihood:

  for (i in 1:M){
    y[i] ~ dbin(pi[i],T)
    pi[i] <- z[i]*p[i]
    z[i] <- step(N-i)
    logit(p[i]) <- z[i]*logitp1[i] + (1-z[i])*logitp2[i]

    logitp1[i] ~ dnorm(alpha,tau)
    logitp2[i] ~ dnorm(alphaprior,tauprior)

   # alpha prior and tauprior are pseudo-prior parameters entered as data

  }
}
```

APPENDIX B: JAGS CODE FOR SECR MODEL

In this appendix we provide sample JAGS code for the SECR models considered in Section 4.2.

B.1. First semi-complete data likelihood approach - SCD1. The model component of the JAGS code is provided here for the semi-complete data likelihood approach using the Jeffreys' prior specification for N.

```
model{
  # Priors:
  sigma ~ dunif(0,10)
  tau <- 1/(sigma*sigma)
  for(i in 1:n){
```

```
    X[i] ~ dunif(xlim[1], xlim[2])
    Y[i] ~ dunif(ylim[1], ylim[2])
  }

  # pdot = probability of being detected at least once (given location)
  # Calculate esa numerically using the integration grid

  for(i in 1:G){ # G = number of points on integration grid
    for(s in 1:S){
      for(k in 1:K){
        one_minus_detprob[i,s,k] <- 1 - exp(-dist2[i,k]*tau/2)
      }
    }
    pdot.temp[i] <- 1 - prod(one_minus_detprob[i,,])
    pdot[i] <- max(pdot.temp[i], 1.0E-10)
  }
  esa <- sum(pdot[])*a # a = size of grid square in numerical integration
  pstar <- esa / A

  # Prior for N: Jeffreys' prior - this is incorporated in the zero trick below
  # in specifying the likelihood term
  # However a prior distribution is needed to be specified on N
  # Use a discrete Uniform prior so the only influence on the posterior
  # distribution is the upper limit

  n00 ~ dcat(prior[]) # prior = rep(1/(M+1-n),M+1-n); entered as data
  n0 <- n00 - 1
  N <- n + n0

  # Zero trick for likelihood component for unobserved individuals
  logzeroprob <- loggam(N) - loggam(n0+1) - loggam(n+1) + n0*log(1-pstar)
  lambda <- -logzeroprob + 1000
  dummy ~ dpois(lambda)  # dummy = 0; entered as data

  # Model for capture histories of observed individuals:
  for(i in 1:n){
    for(k in 1:K){
      for(s in 1:S){
        capthist[i,s,k] ~ dbern(detprob[i,s,k])
        detprob[i,s,k] <- exp(-r2[i,k] * tau/2 )
      }
      r2[i,k] <- pow(X[i] - traps[k,1], 2) + pow(Y[i] - traps[k,2], 2)
    }
  }
}
```

Alternative prior specifications for N can be easily incorporated. Example code for the Negative-Binomial and Beta-Binomial priors follow analogously to those provided in Section A.1.

B.2. Second semi-complete data likelihood approach - SCD2. The model component of the JAGS code is provided here for the semi-complete data likelihood approach, specifying the

posterior conditional distribution of $N - n$ to be of Negative-Binomial form.

```
model{
  # Priors:
  sigma ~ dunif(0,10)
  tau <- 1/(sigma*sigma)
  for(i in 1:n){
    X[i] ~ dunif(xlim[1], xlim[2])
    Y[i] ~ dunif(ylim[1], ylim[2])
  }

  # Posterior conditional distribution for N-n (and hence N):

  n0 ~ dnegbin(pstar,n)
  N <- n + n0

  # pdot = probability of being detected at least once (given location)
  # calculate esa numerically using the integration grid

  for(i in 1:G){ # G = number of points on integration grid
    for(s in 1:S){
      for(k in 1:K){
        one_minus_detprob[i,s,k] <- 1 - exp(-dist2[i,k] * tau/2)
      }
    }
    pdot.temp[i] <- 1 - prod(one_minus_detprob[i,,])
    pdot[i] <- max(pdot.temp[i], 1.0E-10)
  }
  esa <- sum(pdot[])*a # a = size of grid square in numerical integration
  pstar <- esa / A

  # Zero trick for initial 1/pstar^n

  loglikterm <- -n * log(pstar)
  lambda <- -loglikterm + 1000
  dummy ~ dpois(lambda) # dummy = 0; entered as data

  # Model for capture histories of observed individuals:

  for(i in 1:n){
    for(k in 1:K){
      for(s in 1:S){
        capthist[i,s,k] ~ dbern(detprob[i,s,k])
        detprob[i,s,k] <- exp(-r2[i,k] * tau/2 )
      }
      r2[i,k] <- pow(X[i] - traps[k,1], 2) + pow(Y[i] - traps[k,2], 2)
    }
  }
}
```

B.3. Super-population complete data likelihood approach - CD:R. The model component of the JAGS code for the super-population complete data likelihood approach.

```
model {
  # Priors:
  psi ~ dbeta(0.001,1)
  sigma ~ dunif(0,10)
  tau <- 1/(sigma*sigma)
  for(i in 1:M){
    z[i] ~ dbern(psi)
    X[i] ~ dunif(xlim[1], xlim[2])
    Y[i] ~ dunif(ylim[1], ylim[2])
  }

# Complete data likelihood component:

  for(i in 1:M){
    for(k in 1:K){
      for(s in 1:S){
        capthist[i,s,k] ~ dbern(detprob[i,s,k])
        detprob[i,s,k] <- z[i] * exp(-r2[i,k] * tau/2)
      }
      r2[i,k] <- pow(X[i] - traps[k,1], 2) + pow(Y[i] - traps[k,2], 2)
    }
  }

# Calculate N:
  N <- sum(z[])
}
```

B.4. Super-population complete data likelihood approach - CD:DE. The model component of the JAGS code for the super-population complete data likelihood approach.

```
model{

  # Priors:
  psi   ~ dbeta(0.001,1)
  sigma ~ dunif(0,10)
  tau <- 1/(sigma*sigma)

  # Data augmentation part - using Durban and Elston approach:

  for(i in 1:M){

    # Define the first N individuals to be in population of interest

    z[i] <- step(N-i)   # z = 1 if i \le N; z = 0 if i > N.

    # Prior for home range centre for an individual in the population

    X1[i] ~ dunif(xlim[1], xlim[2])
```

```
    Y1[i] ~ dunif(ylim[1], ylim[2])

    # Set pseudo-prior for home range centre for an individual in the population
    # Independent Beta priors for (x,y) location scaled to be in specified region

   Xtemp ~ dbeta(xprior[1],xprior[2]) # xprior - pseudo-prior parameters entered as data
   Ytemp ~ dbeta(yprior[1],yprior[2]) # yprior - pseudo-prior parameters entered as data

   X2[i] <- xlim[1] + Xtemp*xlim[2]
   Y2[i] <- ylim[1] + Ytemp*ylim[2]

   X[i] <- z[i] * X1[i] + (1 - z[i]) * X2[i]
   Y[i] <- z[i] * Y1[i] + (1 - z[i]) * Y2[i]

  }

  # Prior for N: (Jeffrey's prior over {n,n+1,...,M} following Link 2013).

  n00 ~ dcat(prior[])  # prior = rep(1/(M+1-n),M+1-n); entered as data
  n0 <- n00 - 1
  N <- n+n0

  # Use zero trick for factorial term
  # Note loggam(N) instead of loggam(N+1) because of Jeffrey's prior for N

  logLik <- loggam(N) - loggam(n0 + 1) - loggam(n + 1)
  phi <- -logLik + 100000
  dummy ~ dpois(phi)    # dummy = 0; entered as data

  for(i in 1:M){
    for(k in 1:K){
      for(s in 1:S){
        capthist[i,s,k] ~ dbern(detprob[i,s,k])
        detprob[i,s,k] <- z[i] * exp(-r2[i,k] * tau/2)
      }
      r2[i,k] <- pow(X[i] - traps[k,1], 2) + pow(Y[i] - traps[k,2], 2)
    }
  }
}
```

REFERENCES

Durban, J. W. and Elston, D. A. (2005) Mark-recapture with occasion and individual effects: abundance estimation through Bayesian model selection in a fixed dimensional parameter space. *Journal of Agricultural, Biological, and Environmental Statistics* **10**, 291-305

Royle J. A., Dorazio R. M. and Link W. A. (2007) Analysis of multinomial models with unknown index using data augmentaton. *Journal of Computational and Graphical Statistics*, **16**, 67-85

373 Address of King
School of Mathematics, University of Edinburgh,
James Clerk Maxwell Building,
The King's Buildings, Peter Guthrie Tait Road,
Edinburgh, UK. EH9 3FD .

Address of McClintock
National Marine Mammal Laboratory,
Alaska Fisheries Science Center,
NOAA National Marine Fisheries Service,
7600 Sand Point Way NE, Seattle,
Washington 98115 USA

374 Address of Kidney and Borchers
Centre for Research into Ecological and Environmental Modelling
and School of Mathematics and Statistics,
The Observatory, Buchanan Gardens,
University of St Andrews, St Andrews, UK. KY16 9LZ.